State of New York)
: SS:
County of New York)

AFFIDAVIT

On this day, Barbara Gallo personally appeared before me and after being duly sworn, deposes and states:

1. That she is employed as a translator for the language combination German to English by Kenyon & Kenyon (One Broadway, New York, New York 10004);

2. That she has carefully made the attached English language translation from the original document,

German Patent Application 100 00 303.6 filed on January 5, 2000 at the German Patent Office titled:

"Verfahren und Vorrichtung zum Austausch von Daten zwischen wenigstens zwei mit einem Bussystem verbundenen Teilnehmern"

(Method and Device for the Exchange of Data Between at Least Two Users Connected to a Bus System)

written in German; and

3. That the attached translation is a true and correct English version of such original to the best of her knowledge and belief.

Barbara Gallo
BARBARA GALLO

Subscribed and Sworn to before me this 16th day of September, 2003.

Shirley O. Saed
Notary Public

SHIRLEY O. SAED
Notary Public, State of New York
No. 01SA6072143
Qualified in New York County
Certificate Filed in New York County
Commission Expires January 27 2007

METHOD AND DEVICE FOR THE EXCHANGE OF DATA BETWEEN AT LEAST TWO USERS CONNECTED TO A BUS SYSTEM

Background Information

The invention relates to a method and a device for the exchange of data between at least two users **[nodes]** connected to a bus system according to the features of the independent claims.

As related art in automotive networking, the CAN **[controller area network]** protocol has been used for several years. In that case, the communication is controlled in an event-driven manner. Very great loads can be produced when the transmission of various messages is to be initiated at the same time. The non-destructive arbitration mechanism of CAN guarantees the sequential transmission of all messages according to the priority of their identifiers or identifications. For hard real-time systems, an analysis of the transit times[1] and bus loads for the entire system must be made in advance, to make sure that all message deadlines can be observed (even under peak load).

There are already communication protocols which are based on time-controlled processing, such as TTP/C or Interbus-S. The exceptional feature in this case is that the bus access is already planned in advance by allocating transmission instants. Thus, no collisions can occur during the transit time. However, in the same way, a peak load is avoided on the communications bus. Thus, the bus is frequently not completely utilized to capacity.

---

[1] **__Translator's note:__ Another translation of this word is "runtimes".**

When working with such time-controlled systems - systems having distributed clocks - synchronization mechanisms are necessary and known, such as synchronization to level changes of individual bits, etc.

Because of this, intervals are frequently necessary between individual messages to compensate for tolerances, thus reducing the efficiency of the bus capacity utilization.

It is apparent that the related art is unable to deliver optimal results in every respect.

Advantages of the Invention

Thus, the present invention includes a method and a device for the exchange of data in messages between at least two users **[nodes]** which are connected by a bus system and have separate time bases, the messages containing the data being transmitted by the users via the bus system; and a first user, in a function as timer, controls the messages as a function of time in such a way that it repeatedly transmits a reference message, which contains time information regarding the time base of the first user, via the bus at a specifiable time interval; the at least second user forms its own time information, using its time base, as a function of the time information of the first user; a correction value is ascertained from the two pieces of time information; and the second user adapts its time information and/or its time base as a function of the correction value.

In this context, the CAN bus is used particularly expediently as the basic bus system or bus protocol. However, the invention relates generally to any bus system or bus protocol in which an object-oriented message transfer or data transfer is used, thus the

message and/or the data contained therein are clearly recognizable by an identifier. Consequently, this is valid for all busses in which not the users, but rather the messages or their data are addressed, in particular the CAN bus.

The messages are advantageously controlled by a first user as a function of time in such a way that the first user repeatedly transmits a reference message via the bus at at least one predefinable time interval, and the time interval is subdivided into timing windows of specifiable length, the messages being transmitted in the timing windows.

Therefore, compared to the related art, the present invention advantageously includes a higher protocol layer with respect to the actual bus (CAN) protocol which is retained unchanged within the framework of the time-controlled communication according to the invention. Thus, the time-controlled communication advantageously allows full capacity utilization of the bus, and at the same time makes it possible to hold the latency times for each message to a defined value.

The present invention therefore includes a cyclically proceeding transfer of bus (CAN) messages. In this manner, a deterministic and combinable communication system is produced. Such a system is subsequently referred to as TTCAN in this invention. Furthermore, in the same way, the starting point is a CAN bus, the considerations being valid generally for all bus systems or bus protocols having object-oriented message transfer, as mentioned above.

The reference message and the subsequent timing windows up to the next reference message are expediently combined to form a first cycle of specifiable length and/or

specifiable structure, the structure corresponding to the length, number and time position of the timing windows in the time interval which follow the reference message.

Furthermore, a plurality of first cycles of the same structure are advantageously combined to form a second cycle, messages in the second cycle also being repeatedly transmitted in timing windows whose time interval is greater than the time length of the first cycle.

One cyclical message transfer is expediently omitted in at least one timing window of the first or the second cycle. In these initially empty timing windows, it is then possible to transfer arbitrating messages, thus such which do not have to be transmitted cyclically, but rather are available when, for example, certain sequences are concluded.

Each first cycle is advantageously started with a reference message, and the at least second user determines an interval of its time base with respect to the time base of the first user. Consequently, the correction value can expediently be ascertained from the difference of two intervals of the time bases of the at least two users.

In this manner, it is advantageously possible to adjust the accuracy of the local clocks distributed in a TTCAN system, in order to synchronize transmission instants and reception instants more precisely than in the related art.

Another advantage is that, because of this, the clocks in the individual stations can exhibit a greater tolerance of accuracy (cheaper components, particularly oscillators) between the synchronization intervals.

Expediently, the first cycles or base cycles (interval between two reference messages) can become greater, resulting in a rise in efficiency of the bus capacity utilization.

It is also advantageous that the length of a base cycle is no longer limited by the tolerance of the individual clocks, and that the intervals between the individual messages (so-called inter-frame gaps), necessary in other methods to compensate for tolerances, may be omitted.

Further advantages and advantageous refinements come to light from the description and the features of the Claims.

Drawing

In the following, the invention is presented with reference to the Figures contained in the Drawing, in which

Figure 1 shows schematically a bus system having a plurality of users **[nodes]**;

Figure 2 shows the sequence in principle of the first cycles or base cycles and the second cycles, the overall cycles over time;

Figure 3 illustrates in detail the design and message occupancy of the timing windows;

Figure 4 then shows an overall cycle having 7 base cycles and diverse transmission groups of messages, as well as arbitrating messages;

Figure 5 shows, by analogy with Figure 1, the correction (drift correction) of the local time bases or

time information;

Figure 6 shows in detail, in the form of a block diagram, an example for the drift correction itself. This block diagram can be implemented in hardware and/or software.

Description of the Exemplary Embodiments

TTCAN is based essentially on a time-controlled, periodic communication which is clocked by a timer (node, user) with the aid of a time-reference message, or reference message RN for short. The period up to the next reference message RN is denoted as base cycle and is subdivided into n timing windows (see Figure 2). Each timing window allows the exclusive transmission of one periodic message of variable length. These periodic messages are transmitted in a TTCAN controller by using timing marks which are coupled to the expiration of a logical relative time. However, TTCAN also allows the consideration of vacant timing windows. These timing windows can be used for so-called spontaneous messages, the access within these timing windows to the bus being utilized via the arbitration scheme of CAN (arbitrating messages). The synchronization of the timer clock (global time gZ) with the internal local time of individual nodes 1Z1 through 1Z4 is taken into account and efficiently converted.

Figure 1 shows a bus system 100 having a plurality of bus users 101 through 105. In this context, each user 101 through 105 has a separate time base 106 through 110 which can be transmitted, on the one hand, by an internal means such as a clock, counter, clock generator, etc., or externally to the respective user. The respective local time base 1Z1 through 1Z4 is, in particular, a counter, for example, a 16-bit incrementing **[counter]**, which may only be influenced by a HW reset. In this instance, the

local time base is implemented in each node or user 102 through 105. One user, the timer, 101, has an exposed setting. Its time base is designated as global time base 106 having global time gZ, and is either implemented in timer 101, or is transmitted to it externally. Global time gZ is formed in principle in each node from local time base 107 through 110, i.e. local time lZ (lZ1 through lZ4) and an offset OS1 through OS4. As a rule, this offset OSg for timer 101 is equal to zero (OSg = 0). All other nodes form their view of global time gZ from local time lZ (lZ1 through lZ4) and local offset OS1 through OS4 and OSg, when OSg ≠ 0. The case when OSg is not zero occurs, for example, when global time gZ is transmitted from the outside to timer 101 which, in addition, contains its own time base 106. Then, the timer is also calibrated to global time gZ, and gZ and time base 106 may not agree. The local offset is the difference between the local time at the transmission instant (SOF, start of frame) of the reference message, and the global time transmitted by the timer in this reference message.

Local Time Base and the Global Time

Local Time Base: The local time base is a counter, such as a 16-bit incrementing **[counter]**, which may only be influenced by a HW reset. The local time base is implemented in each node.

Reference-Mark Buffer Register: In response to each assumed SOF, the buffer register is loaded with the local time base.

Reference Marker: If the instantaneous message is recognized as a reference message, then the value is retrieved from the buffer register into the reference marker (as local reference mark). The reference marker is

configured, for example, as a 16-bit register.

Timer Reference Mark: This is the reference mark of the timer received by the time takers in the reference message.

Local Offset With Respect To the Global Time: The local offset to the global time is the difference between the reference mark in the buffer register and the global timing mark received in the reference message. It is used for calculating the global time from the local time. The offset of the timer itself remains constant. The timer transmits its local reference mark plus the local offset in the reference message.

Thus, timer 101 is also that node or user which transmits time-reference message 111, i.e., reference message RN for short. Arrow 112 indicates that reference message RN 111 is dispatched, especially simultaneously, to remaining users 102 through 105.

Reference message RN is the basis for the time-controlled, periodic operation of TTCAN. It is clearly denoted by a special identifier, and is received by all nodes, here 102 through 105, as clock generator. In principle, it is sent out cyclically by timer 101. The reference message can include the following data: The number of the instantaneous base cycle BZn, **[and]** the reference mark of the timer in global time.

The reference mark is formed by the takeover of the internal counter reading at the instant of the "start of frame" bit (SOF) upon reception of the reference message of the timer. Thus, the reference mark is an instantaneous recording of the local time base at the instant of receiving the reference message. Relative time RZ1 through RZ4 and RZg specified in the users is the

difference between the local time base and the last reference mark. All definitions with regard to the timing marks used relate to the relative time of an individual user. For example, it can be permanently available as a signal (e.g., by combining the two register values via gates). The reference mark determines the relative time of all nodes on the TTCAN bus.

Watchdog Wg and W1 through W4, likewise shown, is a special relative instant. Such a relative instant (watchdog) - at which a new reference message, and thus a reference mark, as well, is expected at the latest - is defined in each node. Therefore, the watchdog represents a special timing mark. The watchdog is used primarily in the initializing and re-initializing to monitor whether a communication has taken place at all. In this case, the watchdog should always be greater then the interval between the reference messages.

In this context, one timing mark is a relative instant which establishes the relationship between the relative time and an action in the original bus (CAN) controller. One timing mark is represented as **[a]** register, a controller having the ability to manage a plurality of timing marks. A plurality of timing marks <u>can</u> be allocated to one message (see, for example, in Figure 4: transmission group A occurs both in timing window ZF1a, and in timing window ZF4a).

With regard to the application, an application watchdog is actuated in particular. This watchdog must be employed regularly by the application in order to signal the proper operation to the TTCAN controller. The messages are sent by the CAN controller only when this watchdog is actuated.

Figure 2 shows the principle of the time-controlled,

periodic message or data transfer over time. This message transfer is clocked by the timer with the aid of the reference message. Time interval t0 through t6 is designated as base cycle BZ and is subdivided into k timing windows ($k \varepsilon N$). Reference messages RN of respective base cycles BZ0 through BZ3 are transmitted from t0 to t1, t6 to t7, t12 to t13 and t18 to t19, thus in timing window ZFRN. The structure of timing windows ZF1 through ZF5 following a reference message RN, thus their length (in segments S, where Δts = tsb - tsa), their number and their time position, is specifiable. In this manner, an overall cycle GZ1 which begins at t0 and ends at t24, to be run through anew, can be formed from a plurality of base cycles of the same structure. For example, the timing windows include 2 to 5 segments having, for instance, 32 bit times each. The number of timing windows is, for example, 2 to 16, only one timing window or more than 16 also being possible. The number of base cycles in an overall cycle is 2^m, for instance, where, in particular $m \leq 4$.

By way of example, tzff1 and tzff2 denote two transmission enabling intervals or timing-window enabling intervals which, for instance, last 16 or 32 bit times and describe the time frame within which it is possible to begin with the transmission of the message with respect to the base cycle.

Each timing window allows the exclusive transmission of a periodic message of variable length. By way of example, Figure 3 shows two messages of different length and the allocation in the timing window. For instance, message 1 (N1) as block 300 contains 130 bits, and message 2 (N2) as block 301 contains 47 bits.

As already mentioned, maximum and minimum timing windows can be predefined as a function of the message length,

for instance, between 2 and 5 segments per timing window in this example. Thus, a maximum timing window ZFmax is predefined as block 302 which includes 5 segments (S1 through S5) having 32 bit times each, and a minimum timing window ZFmin is predefined as block 303 which includes 2 segments (S1 and S2) having 32 bit times each. Messages N1 and N2 are transmitted in them, the messages thus not having to completely fill up the timing windows; rather, the timing-window sizes are specified according to the message length. Therefore, ZFmax must offer sufficient time or place for the longest possible message, e.g. 130 bits or bit times, and ZFmin can be adapted to the shortest possible message, e.g. 47 bits.

In general, the timing window is the time frame available for a specific message (see Figure 3). The timing window of a message is opened with the application of the proceed-to-send signal, and the beginning of this window agrees in principle with a defined timing mark. The length of the timing window is determined from i segments having, for example, 32 bit times (see block 304a). In this context, the segmenting at, in particular, 32 bit times represents a HW-friendly size. The timing window may not be shorter than the longest message occurring in this timing window. The bit time is, in particular, the nominal CAN bit time.

The transmission enabling interval or timing-window enabling interval describes the time frame within which the transmission of the message may be started. The transmission enabling interval is a part of the timing window. Thus, the proceed-to-send **[signal]** is applied in the interval **[of]** timing mark and timing mark plus delta. The value delta is perceptibly smaller than the length of the timing window (e.g., 16 or 32 bit times for ZFF1 or ZFF2). A message whose beginning does not lie within the

transmission enabling interval may not be sent.

Figure 4 now represents an overall cycle (transmission matrix) GZ2.

Overall Cycle (Transmission Matrix): All messages (RN, A through F and arbitrating) of all users are organized as components of a transmission matrix (see Figure 4). The transmission matrix is made up of individual base cycles BZ0a through BZ7a. All base cycles of overall cycle GZ2 have the same structure. These base cycles can be composed selectively from exclusive (A through F) and arbitrating components. The total number of lines (thus base cycles BZ0a through BZ7a) here is a number $2^m = 8$, where m = 3.

One base cycle (line of the transmission matrix) begins with a reference mark in reference message RN and is composed of a plurality (i) of successive timing windows of defined length (first timing window ZFO, i.e., ZFRN for RN). The arrangement of the messages within the base cycle can be freely established. A timing window is linked for exclusive components to a CAN message object. A timing window can also be left free (409, 421, 441, 417, 445) or be used for arbitrating components (403, 427).

Messages which are always sent in the same timing window but in different base cycles form a transmission group (column of the transmission matrix, A through F) (see Figure 4). Consequently, a period can be established, e.g., A in ZF1a and ZF4a: 401, 407, 413, 419, 425, 431, 437, 443 and 404, 410, 416, 422, 428, 434, 440, 446. One message object (of a timing window) can be sent repeatedly within a transmission group. The period of a message within a transmission group must be a number 2^1, where 1 <= m.

The message object, i.e., the message, corresponds to the message object of the bus, particularly in CAN, and includes the identifier, as well as the data itself. In TTCAN, the message object is supplemented by at least one, preferably by all three, of the following entries in the transmission matrix: timing window, base mark, rate of repetition.

The timing window is the position (ZF0, ZF1a through ZF5a) in the base cycle (BZn, line of the transmission matrix). The beginning of the timing window is defined by reaching a specific timing mark.

The base mark indicates in which base cycle (BZ0a through BZ7a) in the overall cycle the message is sent first. The rate of repetition defines after how many base cycles this transmission is repeated.

To indicate the validity of a message object for the CAN controller, there is a "permanent transmission request" which signifies a permanent enabling of the object (for exclusive components, see below) and an "individual transmission request" which signifies a one-time validity of the object (for arbitrating components, see below).

The automatic retransmission from CAN is advantageously deactivated for the messages in TTCAN.

In the following, the message transfer - periodic messages and spontaneous messages - in the base cycle or in the overall cycle, particularly with respect to the application, is now described again. In so doing, exclusive messages, thus periodic messages, and arbitrating, thus spontaneous messages, are again differentiated.

Exclusive Message Objects (Periodic Messages):

Exclusive message objects are sent when the application watchdog is set, the "permanent transmission request" of the application is set at the CAN controller, and the transmission enabling interval of the appertaining timing window is open. In this case, the timing mark for the message object agrees with the relative time. The permanent transmission request remains set until it is reset by the application itself.

Arbitrating Message Objects (Spontaneous Messages):

Arbitrating message objects are sent when the application watchdog is set, the "individual transmission request" of the application is set at the CAN controller, and the transmission enabling interval of the next timing window specified for this is open. The timing mark for this timing window is then equal to the relative time. The transmission request is reset after successful transmission by the CAN controller. The simultaneous access of various spontaneous messages is regulated via the bit arbitration of CAN. If a spontaneous message in this timing window loses versus another spontaneous message, then it can first fight again for bus access in the next timing window specified for this.

If the entire transmission matrix, i.e. the overall cycle is run through, then a cyclical, time-controlled message transmission is yielded. Time-controlled means that each action uses, as a point of departure, the attainment of a specific instant (see timing marks and relative time). If the overall cycle is run through completely, i.e., all base cycles were processed once, then a start is made again with the first base cycle of the transmission matrix. No time gaps arise in the transition. An overview of such a time-controlled communications system with

timer has been presented in the description and the figures of the Drawing.

By analogy with Figure 1, Figure 5 shows a system having users and a bus system. In this context, a time drift of local times or time bases is determined in a base cycle n beginning at instant tn. The drift correction is then effected in the following base cycle n+1 beginning with tn+1. The times or time bases are then corrected in the next base cycle n+2 beginning with tn+2.

Figure 6 now shows in detail a drift correction according to the present invention. Each node (TTCAN controller) includes:

- a local oscillator which, together with a counter, indicates the local time;
- a reference marker in which the local time is buffered upon reception of a reference message. The reference marker is advantageously designed as a double FIFO;
- a local offset register (double FIFO) which contains the difference between the local reference mark and the timer reference mark;
- an offset difference register which contains the difference of the last two offsets;
- a drift-correction-value register which contains the correction value with which the local timer must be subsequently corrected; **[and]**
- a correction-period register.

In a TTCAN system, each base cycle is started with a reference message in which a time reference is stipulated, and each node determines the interval **[separation]** of its local time with respect to this reference.

A local deviation is calculated from the difference of the intervals of two base cycles and placed in relationship to the total time, from which a basic correction value is calculated for the local time. This basic correction value can be refined, in that for each further base cycle, the difference of the local deviation is also used for the correction.

For simple conversion of the algorithm in hardware or software, the usually necessary quotient formation can be replaced during a correction period (base cycle) by an examination as to how often the correction value is contained in the correction period. This can be done, for example, by a repeated subtraction of the correction value from the correction period in a suitable register, until the residual amount is smaller than the subtrahend. Thereupon, an additional counting pulse is inserted or left out depending on the preceding sign of the correction-value change. The difference between the last two local reference marks is added to the remaining residual amount.

Thus, when working with a steady-state system, a synchronization to +/- one bit time over a base cycle is possible.

The terms time-reference message and reference message are synonymous.

Patent Claims

1. A method for the exchange of data in messages between at least two users **[nodes]** which are connected by a bus system and have separate time bases, the messages containing the data being transmitted by the users via the bus system; and a first user, in a function as timer, controls the messages as a function of time in such a way that it repeatedly transmits a reference message, which contains time information regarding the time base of the first user, via the bus at a specifiable time interval; the at least second user forms its own time information, using its time base, as a function of the time information of the first user; a correction value is ascertained from the two pieces of time information; and the second user adapts its time information and/or its time base as a function of the correction value.

2. The method as recited in Claim 1, characterized in that the time interval is subdivided into timing windows of specifiable length, the messages being transmitted in the timing windows.

3. The method as recited in Claim 1, characterized in that a reference message and the subsequent timing windows up to the next reference message are combined to form a first cycle of specifiable length and/or specifiable structure, the structure corresponding to the length, number and time position of the timing windows in the time interval which follow the reference message.

4. The method as recited in Claims 1 and 3, characterized in that a plurality of first cycles of the same structure are combined to form a second cycle, messages in the second cycle also being repeatedly transmitted in timing windows whose time interval is

greater than the time length of the first cycle.

5. The method as recited in Claims 1 and 3, characterized in that each first cycle is started with a reference message, and the at least second user determines an interval of its time base with respect to the time base of the first user.

6. The method as recited in Claims 1 and 5, characterized in that the correction value is ascertained from the difference of two intervals of the time bases of the at least two users.

7. The method as recited in Claims 1 and 2, characterized in that a cyclical message transfer is omitted in at least one timing window of the first or the second cycle, and arbitrating messages are transmitted in this timing window.

8. A device for the exchange of data in messages between at least two users **[nodes]** which are connected by a bus system and have separate time bases, the messages containing the data being transmitted by the users via the bus system; and a first user, in a function as timer, controls the messages as a function of time in such a way that it repeatedly transmits a reference message, which contains time information regarding the time base of the first user, via the bus at a specifiable time interval; the at least second user forms its own time information, using its time base, as a function of the time information of the first user; a correction value is ascertained from the two pieces of time information; and the second user adapts its time information and/or its time base as a function of the correction value.

9. A device for determining a correction value as recited in one of Claims 1 through 8.

10. A bus system for the exchange of data between at least two users,
characterized in that it is used to carry out a method as recited in one of Claims 1 through 7.

Abstract

A method and device for the exchange of data in messages between at least two users **[nodes]** which are connected by a bus system and have separate time bases, the messages containing the data being transmitted by the users via the bus system; and a first user, in a function as timer, controls the messages as a function of time in such a way that it repeatedly transmits a reference message, which contains time information regarding the time base of the first user, via the bus at a specifiable time interval; the at least second user forms its own time information, using its time base, as a function of the time information of the first user; a correction value is ascertained from the two pieces of time information; and the second user adapts its time information and/or its time base as a function of the correction value.

(Figure 5)

Figure Key

Figure 1:

Zeitmarke	Timing mark
Relative Zeit	Relative time
Offset	Offset
Watchdog	Watchdog
Zeitgeber	Timer
Zeitreferenznachricht	Time-reference message

Figure 2:

Gesamtzyklus	Overall cycle
Basiszyklus	Base cycle
Zeitfenster	Timing window
Zeitreferenznachricht	Time-reference message
Zeitfensterfreigabeintervall	Timing-window enabling interval
Bitzeiten	Bit times
weiter bei Start	Continue at start

Figure 3:

Nachricht	Message
Minimales Zeitfenster	Minimum timing window
Maximales Zeitfenster	Maximum timing window
Bitzeit	Bit time
Zeitfensterfreigabeintervall	Timing-window enabling interval

Figure 4:

Gesamtzyklus	Overall cycle
Basiszyklus	Base cycle
Zeile in der Matrix	Line in the matrix
Startzyklus	Start cycle

Zeitfenster	Timing window
Spalte in der Matrix	Column in the matrix
Zeitreferenznachricht	Time-reference message
Arbitrierend	Arbitrating
Sendegruppen	Transmission groups

Arbitrierend ≙ reservierte Zeitfenster für arbitrierende Nachrichten

Arbitrating ≙ reserved timing windows for arbitrating messages

Figure 5:

Locale Zeit	Local time
Globale Zeit	Global time
Basiszyklus	Base cycle
Zeit-Drift	Time drift
Drift-Korrektur	Drift correction
Zeitgeber	Timer
Referenzbotschaft	Reference message

Figure 6:

Driftkorrektur	Drift correction
Blockschaltbild	Block diagram
Zeitreferenzbotschaft von globalen Zeitgeber	Time-reference message from global timer
Lokale Zeitbasis	Local time base
Zwischen-Register	Buffer register
Zeitgeber-Referenzm.	Timer reference marker
Zeitreferenzbotschaft empfangen	Receive time-reference message
Übernahme	Take-over

Referenzmarker neu	Reference marker neu
Referenzmarker alt	Reference marker old
Lokaler Offset neu	Local offset new
Lokaler Offset alt	Local offset old
Zu globalen Zeit	With respect to global time
Botschaft	Message
Grundkorrektur	Basic correction
Driftkorrekturwertreg.	Drift-correction-value register
Feinkorrektur	Fine correction
Ergebnis	Result
Lokaler Zeitabstand	Local time interval
Zeitreferenzbotschaft	Time-reference message
= Korrekturperiode	= Correction period
Aktueller Driftkorrekturwert	Instantaneous drift-correction value
Vorzeichen	Preceding sign
Taktgenerator	Clock generator
Addition zum Rest	Addition to the residual
Ausgang	Output
Komparator	Comparator
1 Takt Zusätzlich	1 pulse additional
1 Takt unterdrücken	1 pulse suppress